IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re application of: Rahul SRIVASTAVA	Art Unit: 2175
Appl. No.: 10/711,791	Examiner: BASHORE, WILLIAM L
Date Filed: 10/05/2004	Atty. Docket No.: ORCL-006/OID-2004-061-01
For: Reducing Programming Complexity in Applications Interfacing With Parsers for Data elements Represented According to a Markup Languages	

Response to Notification of Non-compliant Appeal Brief

Mail Stop **Appeal Brief - Patents**
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Sir:

Further to the Notice of Appeal filed on 10/07/2008 and the Notification of Non-Compliant Appeal Brief ("Notification") mailed 01/15/2009, Appellants submit this paper under 37 CFR § 41.37. As requested in the Notification, only the updated defective section is included in this paper. In particular, an updated Summary of Claimed Subject Matter section, with separate reference to each of the independent claims 1, 21 and 47 with reference to the specification by page and line number and to the drawings, is included in this paper.

It is not believed that extensions of time or fees for net addition of claims are required, beyond those which may otherwise be provided for in documents accompanying this paper. However, in the event that additional extensions of time are necessary to allow consideration of this paper, then such extensions of time are hereby petitioned under 37 C.F.R. § 1.136(a), and any fees required therefore (including fees for net addition of claims) are hereby authorized to be charged to Deposit Account No.: 20-0674.

5

V. SUMMARY OF CLAIMED SUBJECT MATTER

The present invention is directed to the manner in which applications interface with parsers for data elements represented according to a markup language. Figure 1 of the subject application provides the corresponding context and is reproduced below. The specific support for the claimed elements is provided thereafter.

Independent claim 1 relates to a method of parsing a data file (e.g., XML data file 140 of Figure 1) containing multiple data elements (Figure 4A) according to a markup language (paragraph 4 of the specification). The method of claim 1 expressly recites that the method is implemented in a parser 130.

Parser 130 receives a file identifier of the data file to be parsed with an instruction to parse the data file 140 from an application 110 implemented external to the parser. This feature is further supported by step 210 of Figure 2 and paragraphs 42 and 43 of the specification. By providing the parser external to the application (paragraph 58 of the specification) as shown in Figure 1, several applications can share the same parser.

The parser retrieves a first data element (tag "Books" in line 402) from the specified data file (step 220 of Figure 2 and Paragraph 57 of the specification). The parser then determines a portion identifier as a relative location ("/Books" in row 422 Column 440 of Figure 4B, in general the XPATH of the data element according to XML as noted in paragraph 12) for the first data element. The portion identifier indicates a relative location (last sentence of paragraph 57 of the specification) of the first data element with respect to another data element (paragraph 125) in the data file according to the markup language.

The parser provides the portion identifier and the first data element in association to the application, as shown with respect to step 240 of Figure 2 and described in paragraphs 46 and 47 of the specification. Paragraph 47 of the specification further discloses the manner in which the data element and the portion identifier are provided in association for event based parser techniques (XPath provided as a parameter while providing corresponding data element) and object based parsing technique (Xpath

expression provided in the data structure representing XML data file 140 in a random access memory).

By having the parser provide such data element and the portion identifier in association, the programming complexity of applications interfacing with parsers can be reduced (paragraph 30 of the specification).

Claim 3 recites that the parser provides the portion identifier and the corresponding data element according to an application programming interface (API). The API is used by applications to obtain the portion identifier and the data element. The recitations are supported by the API shown in Figures 8A-8F in case of event based push parser.

Claim 4 recites that the API is defined to provide the portion identifier and the first data element as respective parameters of a "single" procedure call. This feature is supported by each of rows 842 and 843 of Figure 8C in that each row shows a respective procedure call with xpath and xpathVal as parameters. The corresponding description is provided in paragraph 96 of the specification.

Claim 5 recites that the parser is an event-based parser, which is defined in paragraphs 7-9 of the specification.

Claim 6 recites that the parsing is performed by an object oriented parser (paragraph 10 of the specification). The parser is recited as storing the first data element and the portion identifier in a data structure (DOM, Paragraphs 10, 67 and 68), with application obtaining the first data element (line 587 of Figure 5B) and the corresponding portion identifier (line 555 of Figure 5B) from the data structure. Paragraphs 70 and 71 provide the corresponding description.

Claim 10 recites that parsing is performed according to a push parsing approach (paragraph 8 of the specification), wherein the API provides a procedure for the parsing to provide the Xpath to the application. This feature is supported by each of rows 842

and 843 of Figure 8C, in that each row shows a respective procedure call with xpath as a parameter.

Independent claims 12 recites a method of implementing an application (e.g., 110 of Figure 1) using data contained in a data file 140. The data file is recited as containing multiple data elements (Figure 4A) according to a markup language (e.g., XML as described in paragraph 5).

The application is recited as instructing a parser to parse the data file of interest, with the parser being implemented external to the application (step 310 of Figure 3 and paragraph 51 of the specification). This feature is further supported by line 520 of Figure 5A and associated description in paragraph 66 in case of Object oriented parser, line 616 of Figure 6 and associated description in paragraph 79 in case of push based (SAX) event parser with corresponding extensions according to aspects in other claims, lines 917 and 919 of Figure 9 and associated description in paragraphs 128/129 in case of a push based event parser API, and lines 708-710 of Figure 7 and the description in paragraph 87 of the specification.

The application is recited as obtaining in association both a portion identifier and a data element from the parser, with the portion identifier indicating a relative location of the data element with respect to another data element in the data file according to the markup language (step 325 of Figure 3 and paragraph 52 of the specification). This feature is supported by lines 555 (portion identifier) and 587 (data element) of Figure 5B and associated description in paragraph 71 in case of Object oriented parser, lines 621, 629 and 637 of Figure 6 and associated description in paragraph 80 in case of push based (SAX) event parser with corresponding extensions according to aspects in other claims, lines 921 and 927 of Figure 9 and associated description in paragraphs 130 in case of a push based event parser API, and lines 713, 716 and 719 of Figure 7 and the description in paragraph 88 of the specification.

The application then processes the obtained data element and the portion identifier (step 330 of Figure 3 and the description of Paragraph 53 of the specification). This

feature is further supported by the last sentence of paragraphs 79 and 88, which indicates the processing can be any business logic as desired by a programmer of the application.

By receiving such data element and the portion identifier in association, the programming complexity of applications can be reduced (paragraph 30 of the specification).

Independent claim 21 relates to a computer readable medium (1130/1140 of Figure 11) causing a digital processing system (1100 of Figure 11) to parse a data file (e.g., XML data file 140 of Figure 1) containing multiple data elements (Figure 4A) according to a markup language (paragraph 4 of the specification).

Claims 21 recites that a file identifier of the data file to be parsed is received from an application 110. This feature is further supported by step 210 of Figure 2 and paragraphs 42 and 43 of the specification.

A first data element (tag "Books" in line 402) is retrieved from the specified data file (step 220 of Figure 2 and Paragraph 57 of the specification). The portion identifier of the retrieved first data element is then set to equal a relative location ("/Books" in row 422 Column 440 of Figure 4B, in general the XPATH of the data element according to XML as noted in paragraph 12) for the first data element with respect to another data element (paragraph 125) in the data file according to the markup language.

The portion identifier and the first data element is provided in association to the application, as shown with respect to step 240 of Figure 2 and described in paragraphs 46 and 47 of the specification. Paragraph 47 of the specification further discloses the manner in which the data element and the portion identifier are provided in association for event based parser techniques (XPath provided as a parameter while providing corresponding data element) and object based parsing technique (Xpath expression provided in the data structure representing XML data file 140 in a random access memory).

Independent claim 47 relates to an article of manufacture for parsing a data file (e.g., XML data file 140 of Figure 1) containing multiple data elements (Figure 4A) according to a markup language (paragraph 4 of the specification).

The article of manufacture is recited to contain:

1. a means for receiving (path 112 of Figure 1, CPU 1100/network interface 1180 of Figure 11) a file identifier of the data file to be parsed from an application 110 (step 210 of Figure 2 and paragraphs 42 and 43 of the specification),

2. a means for retrieving (path 132 of Figure 1, CPU 1100/network interface 1180 of Figure 11) a first data element (tag "Books" in line 402) from the specified data file (step 220 of Figure 2 and Paragraph 57 of the specification), wherein the first data element is presently stored in the data file prior to receiving the file identifier (Figure 4A and associated description in paragraph 56 of the specification, in combination with XML data file 140 of Figure 1),

3. a means for determining (parser 130 of Figure 1, CPU 1100 of Figure 11) a portion identifier of the first data element, the portion identifier indicating a relative location ("/Books" in row 422 Column 440 of Figure 4B, in general the XPATH of the data element according to XML as noted in paragraph 12) for the first data element with respect to another data element (paragraph 125) in the data file according to the markup language, and

4. a means for providing (path 112 of Figure 1, CPU 1100/network interface 1180 of Figure 11) in association the portion identifier and the first data element to the application, as shown with respect to step 240 of Figure 2 and described in paragraphs 46 and 47 of the specification. Paragraph 47 of the specification further discloses the manner in which the data element and the portion identifier are provided in association for event based parser techniques (XPath provided as a parameter while providing corresponding data element) and object based parsing technique (Xpath expression provided in the data structure representing XML data file 140 in a random access memory).

Independent claim 49 recites an application (110 in Figure 1) and an event based parser (parser 130 in combination with paragraphs 7-9 of the specification). The

application instructs the event based parser to parse a data file of interest (step 310 of Figure 3). In response, parser 130 retrieves data from the data file (step 230 of Figure 2), sets a portion identifier to a relative location ("/Books" in row 422 Column 440 of Figure 4B, in general the XPATH of the data element according to XML as noted in paragraph 12), and then provides to the application in association the data element and the portion identifier (step 240 of Figure 2) as a sequence of events without receiving a request (paragraphs 7-9 of the specification; lines 621, 629 and 637 of Figure 6 and associated description in paragraph 80 in case of push based (SAX) event parser with corresponding extensions according to aspects in other claims, lines 921 and 927 of Figure 9 and associated description in paragraphs 130 in case of a push based event parser API).

Accordingly all noted defects are believed to be overcome and continuation of examination is respectfully requested. The Office is invited to telephone the undersigned representative at 707.356.4172 if it is believed that an interview might be useful for any reason.

Respectfully submitted,
/Narendra Reddy Thappeta/
Signature
Date: February 10 2009
Printed Name: Narendra Reddy Thappeta
Attorney for Applicant
Registration Number: 41,416